IN THE CLAIMS

Please **amend** Claims 1 and 5-6, **cancel** Claims 4 and 7-16, and **add** Claims 17-27 as indicated:

1. (currently amended) A method ~~of executing a software application, comprising the steps of~~ comprising:

[[(a)]] calling the software application residing on a server from one of a plurality of clients, the clients and the server connected to each other through at least one network, the software application having a plurality of policy frameworks, each of the frameworks being associated with a respective one of the plurality of clients;

[[(b)]] launching a container/desktop of one of the plurality of clients consistent with the respective policy framework of the one client, wherein the container/desktop includes a software for displaying a user-interface on a display in a computer;

[[(c)]] the container/desktop initializing and communicating to the server to execute a script of the application;

[[(d)]] executing the script on the server, the script downloading a first user-interface component of the application to the container/desktop;

[[(e)]] the container/desktop executing the first user-interface component;

[[(f)]] the first user interface component linking to and starting a subsequent user-interface component of the script; and

[[(g)]] in response to the subsequent user-interface component of the script being started, the software in the container/desktop automatically closing the first user-interface component and removing the first user-interface from a system memory in the computer. [[and]]

~~(h) the server downloading the subsequent user interface component to the container/desktop, and the container/desktop executing the subsequent user interface component and then closing the subsequent user interface component.~~

2. (original) The method of claim 1, further comprising the step of said script starting and executing the user-interface components within a policy framework of the container/desktop.

3. (original) The method of claim 1, further comprising the container/desktop removing the user-interface components from memory within the client when the user-interface component is closed.

4. (cancelled)

5. (currently amended) A computer server, comprising:

(a) a processor, a memory, a bus, and at least one I/O port by which to communicate with a remote client having a container/desktop, wherein the container/desktop includes a software for displaying a user-interface on a display in a computer;

(b) an operating system with which to coordinate the processor, the memory, the bus and the at least one I/O port to communicate to the client;

(c) an application stored in memory of the server;

(d) a script of the application stored in the memory of the server; and

(e) a plurality of user-interface components stored in the memory, the script comprising code to connect the user-interface components to comprise the application;

wherein the application launches the container/desktop on the client which in turn interacts with the script on the server to download each of the user-interface components from the server to the container/desktop on an as-needed basis, and wherein, in response to a subsequent user-interface component of the script being started, the software in the container/desktop automatically closes a previous user-interface component and removes the previous user-interface from a system memory in the computer.

6. (currently amended) A client device, comprising:

(a) a container/desktop, wherein the container/desktop includes a software for displaying a user-interface on a display in a computer;

(b) an I/O port with which to communicate to one or more servers having software applications, scripts, and user-interface components; and

(c) an interactive medium with which to interact with a user, wherein when the user uses the interactive medium to request an application from the server, the container/desktop communicates with the server through the I/O port and invokes a script of the application in the server which downloads user-interface components to the container/desktop according to the script and only on an as-needed basis, and wherein the container/desktop discards the user-interface components no longer needed by the application by removing the no longer needed user-interface components from a system memory in the client device.

7-16. (cancelled)

17. (new) The method of claim 1, wherein the first user-interface component directly passes data to the subsequent user-interface component before the first user-interface component closes.

18. (new) The method of claim 1, wherein the first and subsequent user-interface components are decoupled from the software application, such that an execution context of the user-interface components can be changed without affecting application code in the software application.

19. (new) The method of claim 18, wherein the user-interface components are decoupled via a script on a server managing a contract between the script and a policy of the container/desktop.

20. (new) The method of claim 19, wherein the policy describes a number of tasks that can be simultaneously executed on a client computer.

21. (new) The method of claim 19, wherein the policy describes a visual policy on a client computer, and wherein the visual policy describes a position, sizing and cropping of a user-interface component.

22. (new) A computer-usable medium embodying computer program code, the computer program code comprising computer executable instructions configured to:

call the software application residing on a server from one of a plurality of clients, the clients and the server connected to each other through at least one network, the software application having a plurality of policy frameworks, each of the frameworks being associated with a respective one of the plurality of clients;

launch a container/desktop of one of the plurality of clients consistent with the respective policy framework of the one client, wherein the container/desktop includes a software for displaying a user-interface on a display in a computer;

use the container/desktop to initialize and communicate to the server to execute a script of the application;

execute the script on the server, the script downloading a first user-interface component of the application to the container/desktop;

use the container/desktop to execute the first user-interface component;

use the first user interface component to link to and start a subsequent user-interface component of the script;

in response to the subsequent user-interface component of the script being started, the software in the container/desktop automatically closes the first user-interface component and removes the first user-interface from a system memory in the computer; and

download the subsequent user-interface component to the container/desktop, and the container/desktop executes the subsequent user-interface component and then closes the subsequent user-interface component.

23. (new) The computer-usable medium of claim 22, wherein the first user-interface component passes data to the subsequent user-interface component before the first user-interface component closes.

24. (new) The computer-usable medium of claim 22, wherein the first and subsequent user-interface components are decoupled from the software application, such that an execution context of the user-interface components can be changed without affecting application code in the software application.

25. (new) The computer-usable medium of claim 24, wherein the user-interface components are decoupled via a script on a server managing a contract between the script and a policy of the container/desktop.

26. (new) The computer-usable medium of claim 25, wherein the policy describes a number of tasks that can be simultaneously executed on a client computer.

27. (new) The computer-usable medium of claim 25, wherein the policy describes a visual policy on a client computer, and wherein the visual policy describes a position, sizing and cropping of a user-interface component.